

```
In [22]: #Demonstrate fibonacci using recursion
def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))

nterms = 10

# check if the number of terms is valid
if nterms <= 0:
    print("Plese enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

Fibonacci sequence:

0
1
1
2
3
5
8
13
21
34

```
In [21]: dict={"Id":41,
              "name":"sapna",
              "join_date":"28/04/2024",
              "job_description":"frontend developer",
              "salary":99000
            }
print(dict)

def display_full_name(name):
    parts=name.split()
    prefix=parts[0]
    first_name=parts[1]
    last_name=parts[-1]
    return(prefix , first_name, last_name)

employee=[]
employee.append({"empid":1, "name":" Sapna B","join_date":"02/07/2024","salary":97000})
employee.append({"empid":2, "name":"Akshata K","join_date":"05/07/2024","salary":87000})
employee.append({"empid":3, "name":"Jyoti P","join_date":"07/07/2024","salary":72000})
employee.append({"empid":4, "name":"Shivani J","join_date":"09/07/2024","salary":60000})
employee.append({"empid":5, "name":"Sushmita J","join_date":"11/07/2024","salary":50000})
print(employee)

#senior most employee
senior_employee=max(employee,key=lambda emp: emp["join_date"])
print("\n\nSenior most employee: ",display_full_name(senior_employee["name"]))

#employee added
employee.append({"empid":6, "name":"Deppa H","join_date":"13/07/2024","salary":91000})
print("new employee is added")
```

```

print(employee)

#employee removed
removed_employee= employee[2]
print("Removed Employee: ",display_full_name(removed_employee["name"]))

#sorting
employee.sort(key=lambda emp: emp["name"])

#display full name
print("\n Employee list: ")
for emp in employee:
    print(display_full_name(emp["name"]))

#salary is above 80000
print("The Employee whose salary is greater than 80000: ")
for emp in employee:
    if emp["salary"]>80000:
        print(display_full_name(emp["name"]))

```

```

{'Id': 41, 'name': 'sapna', 'join_date': '28/04/2024', 'job description': 'fronten
d developer', 'salary': 99000}
[{'empid': 1, 'name': ' Sapna B', 'join_date': '02/07/2024', 'salary': 97000}, {'e
mpid': 2, 'name': 'Akshata K', 'join_date': '05/07/2024', 'salary': 87000}, {'empi
d': 3, 'name': 'Jyoti P', 'join_date': '07/07/2024', 'salary': 72000}, {'empid':
4, 'name': 'Shivani J', 'join_date': '09/07/2024', 'salary': 66000}, {'empid': 5,
'name': 'Sushmita J', 'join_date': '11/07/2024', 'salary': 83000}]

```

Senior most employee: ('Sushmita', 'J', 'J')

new employee is added

```

[{'empid': 1, 'name': ' Sapna B', 'join_date': '02/07/2024', 'salary': 97000}, {'e
mpid': 2, 'name': 'Akshata K', 'join_date': '05/07/2024', 'salary': 87000}, {'empi
d': 3, 'name': 'Jyoti P', 'join_date': '07/07/2024', 'salary': 72000}, {'empid':
4, 'name': 'Shivani J', 'join_date': '09/07/2024', 'salary': 66000}, {'empid': 5,
'name': 'Sushmita J', 'join_date': '11/07/2024', 'salary': 83000}, {'empid': 6, 'n
ame': 'Deppa H', 'join_date': '13/07/2024', 'salary': 91000}]

```

Removed Employee: ('Jyoti', 'P', 'P')

Employee list:

```

('Sapna', 'B', 'B')
('Akshata', 'K', 'K')
('Deppa', 'H', 'H')
('Jyoti', 'P', 'P')
('Shivani', 'J', 'J')
('Sushmita', 'J', 'J')

```

The Employee whose salary is greater than 80000:

```

('Sapna', 'B', 'B')
('Akshata', 'K', 'K')
('Deppa', 'H', 'H')
('Sushmita', 'J', 'J')

```